如何用自然语言进行 Google 搜索

Minja　05月27日

如果你经常处理大宗任务，全程键盘输入的操作连贯性会比鼠标更好；在键盘录入中，**自然语言**的使用又可以进一步提升操作速度。很多工具都对自然语言有了很好的支持，有些甚至凭此在同类工具中脱颖而出：

- **日程管理领域**，Fantastical 以自然语言添加日程为傲，打一行字就能设置日程的标题、日期、优先级等属性。
- **任务管理领域**，Todoist 支持通过自然语言记录任务，经典用法是一次记录、分配给全团队的各个人员。
- **文件搜索领域**，Spotlight 特色是可以用自然语言进行文件过滤，比如输入 `昨天下载的文件` 来找文件。

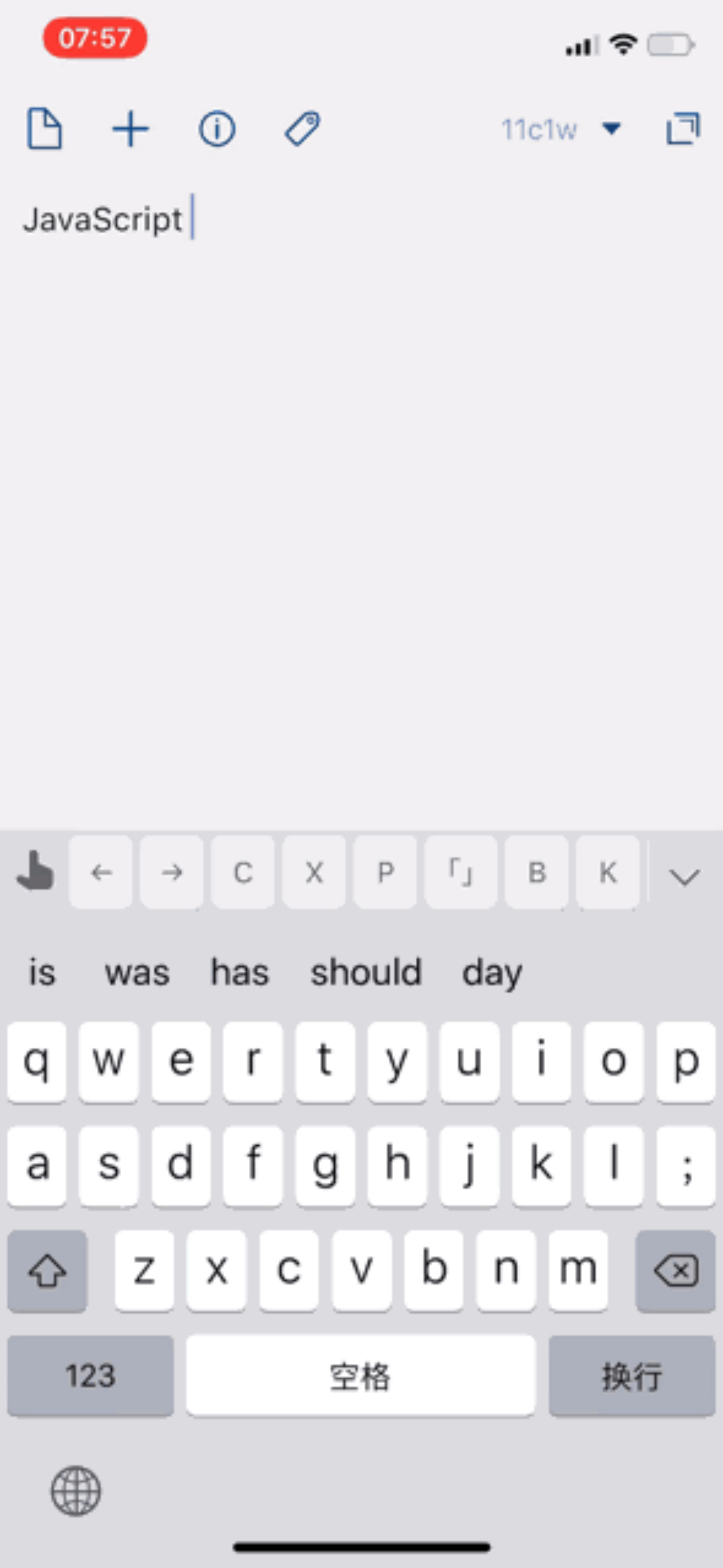
不过，**搜索引擎**这个和我们朝夕相处的功能似乎很少得到自然语言的青睐[1]

，许多搜索设置仍需要在图形菜单里点击。如果你想找一篇 2015 年前后的新闻，还得在工具菜单里面一路点到 4 年前（手机干脆不支持）。

相信我们的读者多数都想兼顾**搜索操作的速度**和**搜索结果的精度**，为此我制作了一套自然语言搜索动作，只需要在搜索关键词后跟上简单后缀，就能**限定 Google 搜索结果的距今**

时限、日期范围或所用语言。动作可以无缝集成在各种 Mac 自动化工具上，也可以在 iOS 的经典文本处理工具 Drafts 中使用[2]

。



JavaScript

07:57
11c1w
is
was
has
should
day
123
空格
换行

在 Drafts 中使用自然语义进行搜索

- [Drafts 动作下载 🔗]
- [Alfred 动作下载 🔗]

- LaunchBar 动作下载 🔗

下面我们先了解一下语法和动作的使用，日常使用的话看完这部分就差不多；随后会用两节内容分析一下动作的设计思路，需要自定义的读者可以从中找到捷径。

语法规则

虽然美名其曰「自然语言」，实际上自动化工具并没有真的「听懂」你的话[3]

，我们还是要遵循一些语法规则。好在这些规则不多，也容易用。由于 iOS 上的 Drafts 版不需要任何设置就能使用，这节就以它为例进行语法展示。

如文章开头所展示的一样，**把搜索关键词和自然语言语法记录在 Drafts 草稿的同一行中**，然后运行「Google it with natural language」动作即可按条件进行搜索。

针对搜索中最常见的几类过滤条件，我设计了这几组语法参数：

1. **设置距今时限**：限制搜索结果的时间为距今多久，找新闻时经常用到，比如限定搜索最近6 小时 h6，就能避免明日黄花混进搜索结果。具体可选这些参数：
   1. **限定年份**：yn，n 表示具体要**几年**，比如最近两年就用 y2，只搜最近一年的话则不填 n，单用一个 y 即可。下面几项形式也一样。

    2. **限定月份**：`mn`。
    3. **限定周数**：`wn`。
    4. **限定天数**：`dn`。
    5. **限定小时**：`hn`。
2. **设置日期范围**：比上一组更加精确的时间条件，可以具体到两个日期之间[4] 。使用空格隔开的两个短日期作为参数即可，比如搜今年四月一日到五月一日期间就是：2019-04-01 2019-05-01
3. **设置结果语言**：限制结果，适合滤掉看不懂的内容，比如日文和中文常出现字同义不同的情况，我就限定只看自己需要的 `中文` 结果。动作默认只设置了 `中文` 和 `英文`，需要其他语言的话看后两节介绍自行添加。

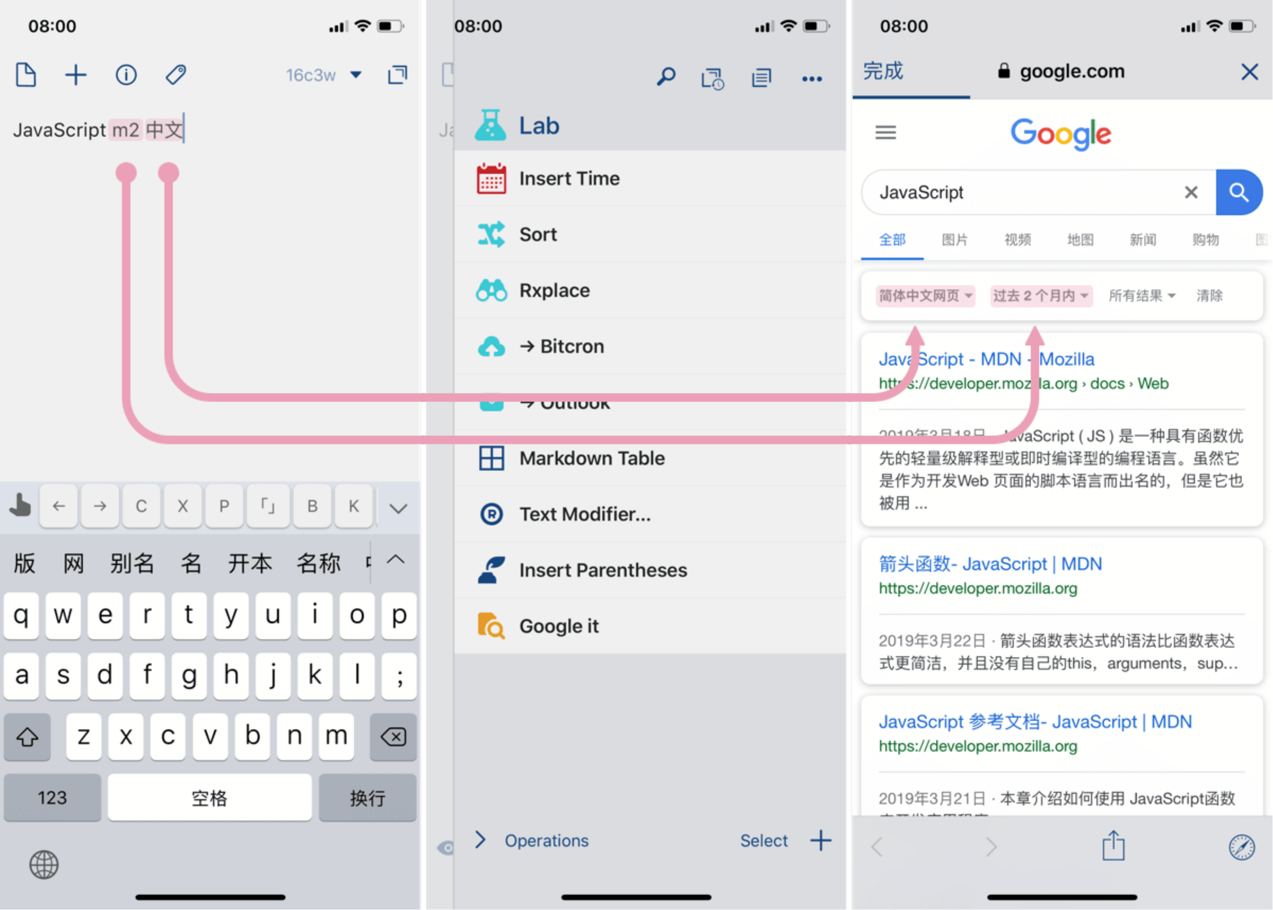
Google 搜索的过滤条件当然不止这些，不过常见的文件类型、所在站点都可以直接当关键词输入，没必要多此一举为它们设计自然语言语法。

注意，**每组语法参数之间需要用空格 ` ` 隔开**，搜索关键词和语法之间同样也要隔开。比较推荐的写法是先写完所有关键词，空一格后再开始加上自然语言语法参数。尽管脚本也能成功识别混在一起的内容，但总有我没考虑到的特殊文本组合，还是按规矩出牌比较保险。

下面来看两个例子，相信看完后对于搜索语法就没什么疑问了。

1. **只用动作自带的语法**：搜索近两个月有关 `JavaScript` 的中文结果：

```
JavaScript m2 中文
```


08:00
16c3w
JavaScript m2 中文
版 网 别名 名 开本 名称
123
空格
换行
Lab
Insert Time
Sort
Rxplace
→ Bitcron
→ Outlook
Markdown Table
Text Modifier...
Insert Parentheses
Google it
Operations
Select
完成
google.com
Google
JavaScript
全部
图片
视频
地图
新闻
购物
简体中文网页
过去 2 个月内
所有结果
清除

JavaScript - MDN - Mozilla
https://developer.mozilla.org › docs › Web
2019年3月18日 · JavaScript (JS) 是一种具有函数优先的轻量级解释型或即时编译型的编程语言。虽然它是作为开发Web 页面的脚本语言而出名的，但是它也被用 ...

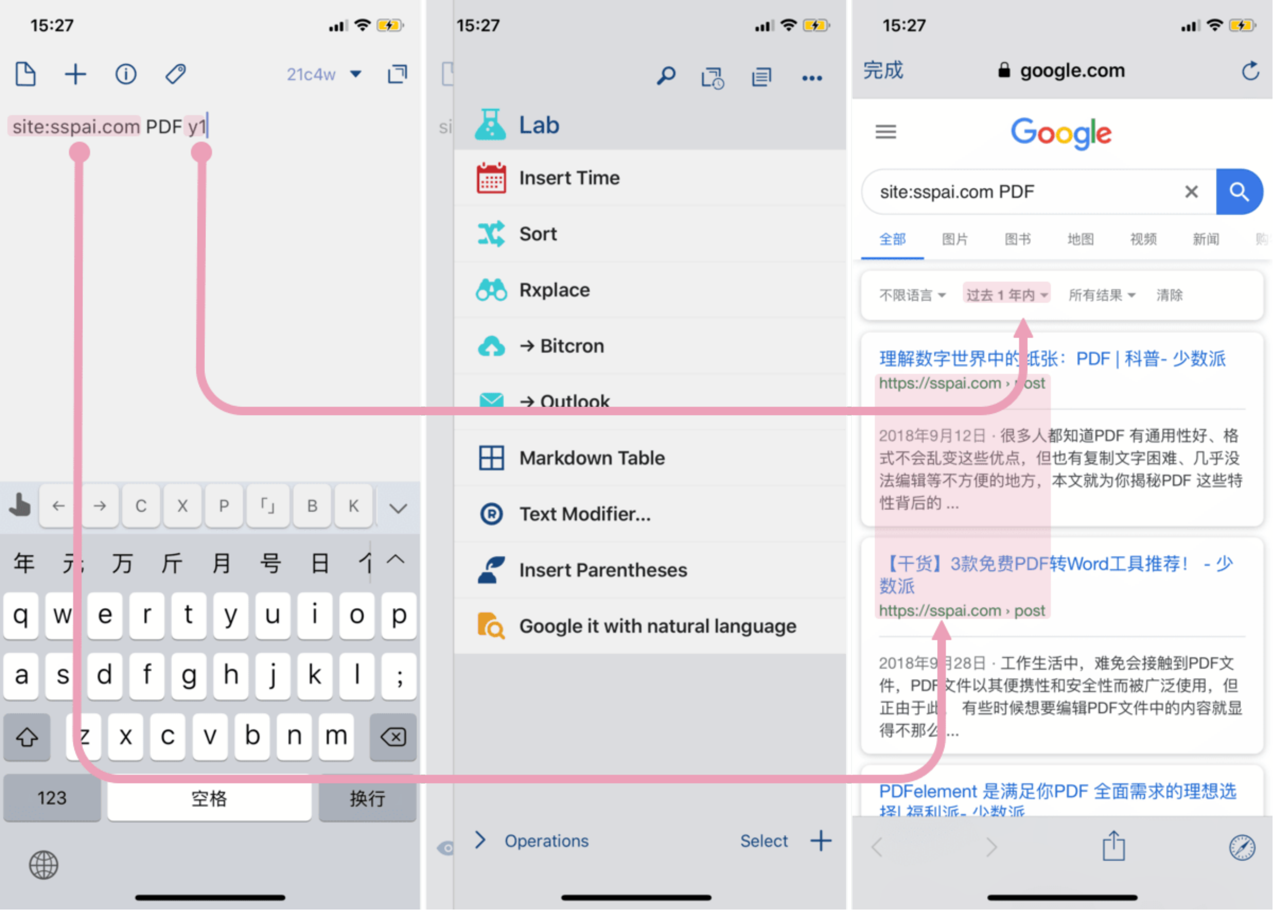
箭头函数- JavaScript | MDN
https://developer.mozilla.org
2019年3月22日 · 箭头函数表达式的语法比函数表达式更简洁，并且没有自己的this，arguments，sup...

JavaScript 参考文档- JavaScript | MDN
https://developer.mozilla.org
2019年3月21日 · 本章介绍如何使用 JavaScript函数

搜索参数对应的搜索结果设置

2. **结合 Google 自己的搜索语法**：搜索近一年，少数派所发布关于 PDF 的文章。其中，`site:sspai.com` 用了 Google 原生语法，表示仅搜索少数派的内容

```
site:sspai.com PDF y1
```


15:27
21c4w
Lab
Insert Time
Sort
Rxplace
→ Bitcron
→ Outlook
Markdown Table
Text Modifier...
Insert Parentheses
Google it with natural language
Operations
Select
完成
google.com
Google
全部
图片
图书
地图
视频
新闻
不限语言
过去 1 年内
所有结果
清除
空格
换行
123

site:sspai.com PDF y1

site:sspai.com PDF

理解数字世界中的纸张：PDF | 科普- 少数派
https://sspai.com › post
2018年9月12日 · 很多人都知道PDF 有通用性好、格式不会乱变这些优点，但也有复制文字困难、几乎没法编辑等不方便的地方，本文就为你揭秘PDF 这些特性背后的 ...

【干货】3款免费PDF转Word工具推荐！ - 少数派
https://sspai.com › post
2018年9月28日 · 工作生活中，难免会接触到PDF文件，PDF文件以其便携性和安全性而被广泛使用，但正由于此，有些时候想要编辑PDF文件中的内容就显得不那么 ...

PDFelement 是满足你PDF 全面需求的理想选择| 福利派- 少数派

结合了 Google 自带语法的四然语言搜索

只要掌握 关键词+自然语言参数 的组合，再配合上 Google 自带的几条语法，就可以在保持搜索的精度的同时大大提升操作流畅度。Google 语法可参我以前写过的《如何提高搜索的精度与速度》一文，没订阅过 Power+ 1.0 的读者也可以在网上搜到网友整理的语法，不一定要为此破费。

iOS 端实现：基于 Drafts

说起来是「自然语言」，实现起来主要就是检查有没有输入特定参数，这事儿交给 JavaScript 比较合适，而且写好的代码在 iOS 和 macOS 上都可以用，少走回头路。

在 iOS 运行 JavaScript 的理想选择是 Drafts，而且它的界面

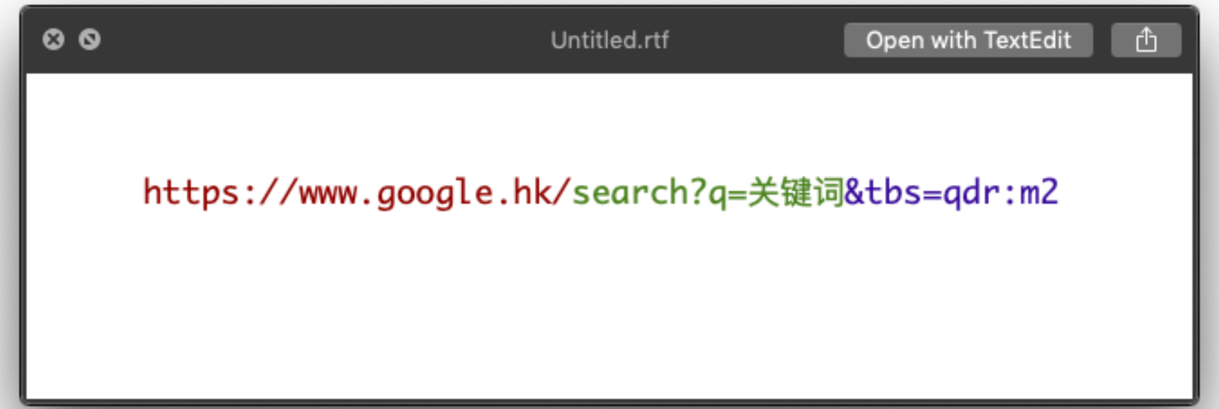
很干净，也适合输入大串关键词和参数。成品搜索动作由**一段 JavaScript 代码**和**一个打开 URL 的步骤**组成，除了头尾涉及和 Drafts 交互的少量代码，JavaScript 脚本核心部分和 Mac 版是通用的。

这一节我们先简单看一下代码的整体思路，然后对于其中可以自定义的部分进行专门说明，最后再分享一个 Drafts 的自动化动作模板套件，方便有能力自己折腾的读者。

整体思路

脚本头尾是拿来和 Drafts 交互数据的，稍后介绍 Drafts 模板时专门讲，现在先来看实现自然语言搜索的部分。

核心代码所做的事情，就是把我们输入的简单参数**「翻译」**成 Google 能读懂的 URL 参数。我们所谓的自然语言只是对于使用者而言，而最终体现在 Google 的搜索 URL 中的则是一串规整却对普通人不怎么友好的参数——比如我们想搜最近两个月的内容，输入的是 `m2`，翻译到 URL 中则是 `&tbs=qdr:m2`。



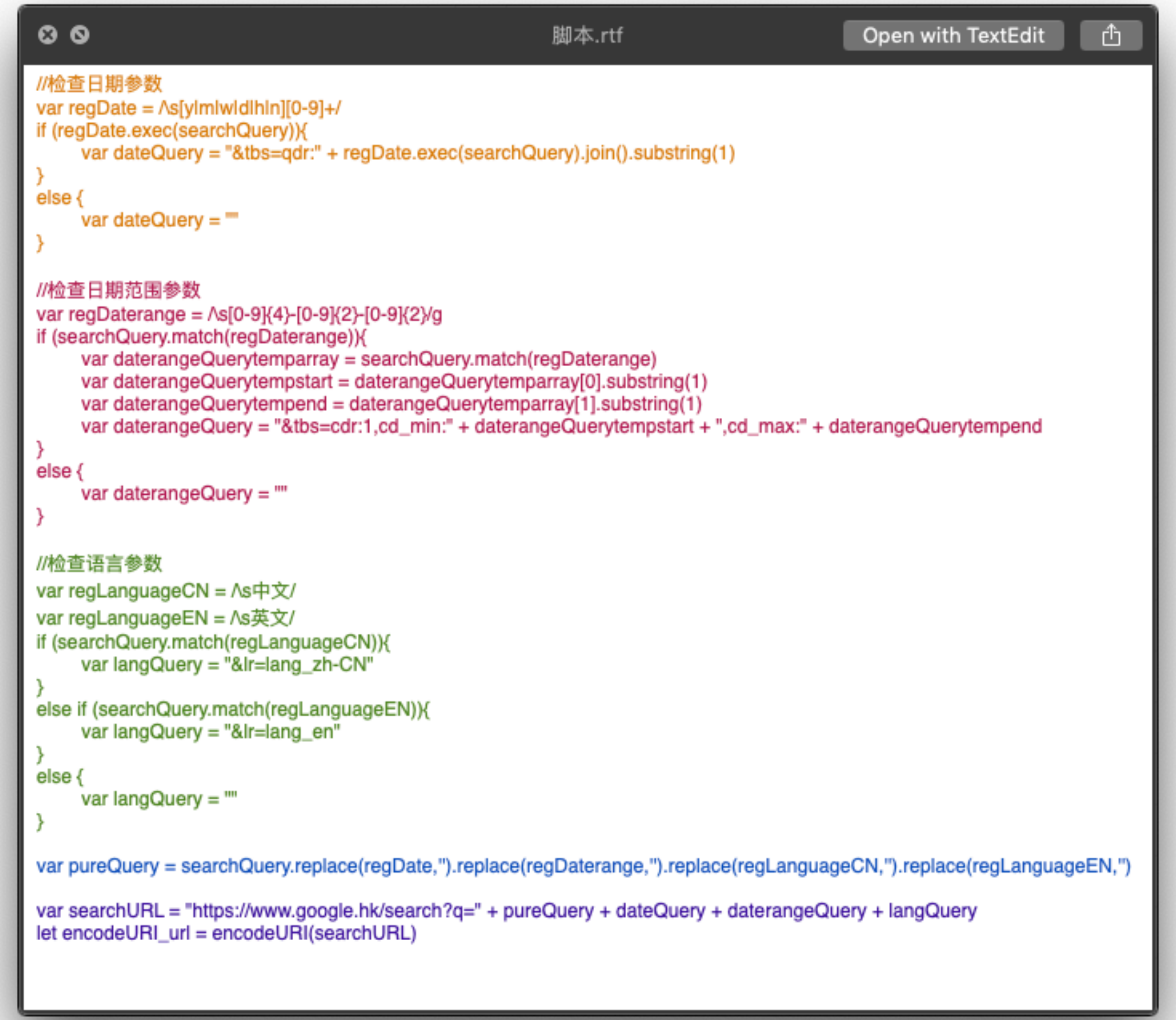
https://www.google.hk/search?q=关键词&tbs=qdr:m2

Google 搜索 URL 的组成

除了简单的日期，还有详细日期范围、结果语言等参数，也可以交给 JavaScript 脚本来「翻译」。总体上，脚本所做的事就是**不断检查有没有符合格式的自然语言参数，然后译为 Google 可以理解的形式。**

1. 橙色部分：检查有没有输入日期自然语言参数，有的话就翻译成日期搜索 URL 参数。
2. 粉色部分：检查有没有输入日期范围参数，并翻译
3. 绿色部分：检测有无结果语言参数，并翻译
4. 蓝色部分：把参数从输入中剔除，剩余部分作为关键词
5. 紫色部分：组合成 `搜索 URL+关键词+搜索参数` URL 形式。为了照顾特殊符号和东亚字符，后续还有一步 URL 编码（encode）的操作。

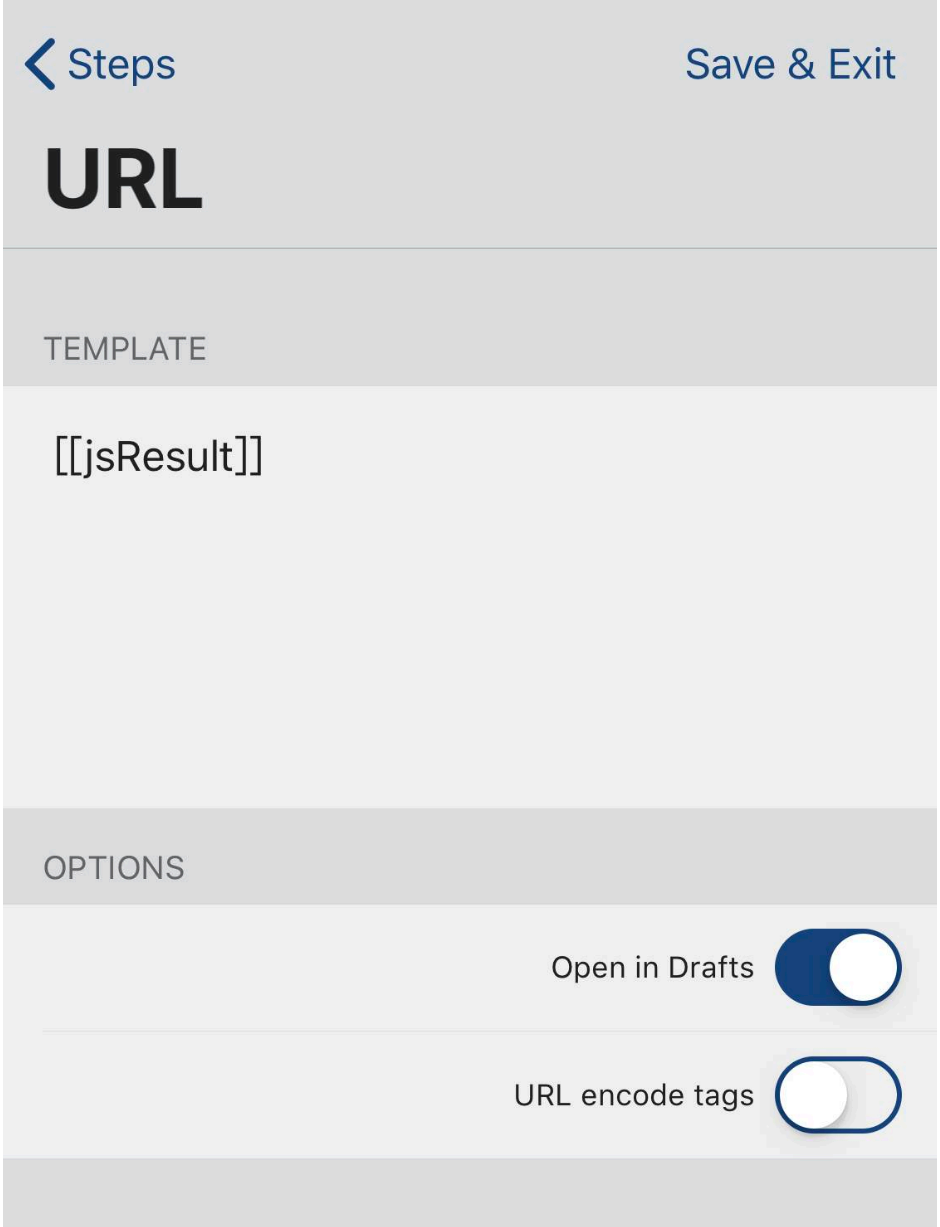
注：粉色部分表示日期参数范围的代码随时会随着 Google 的动作进行微调，使用时不会有任何影响。之后 Google 有更新语法时，本文动作也会在 GitHub 上尽快发布更新。



```
//检查日期参数
var regDate = /\s[y|m|w|d|h|n][0-9]+/
if (regDate.exec(searchQuery)){
        var dateQuery = "&tbs=qdr:" + regDate.exec(searchQuery).join().substring(1)
}
else {
        var dateQuery = ""
}

//检查日期范围参数
var regDaterange = /\s[0-9]{4}-[0-9]{2}-[0-9]{2}/g
if (searchQuery.match(regDaterange)){
        var daterangeQuerytemparray = searchQuery.match(regDaterange)
        var daterangeQuerytempstart = daterangeQuerytemparray[0].substring(1)
        var daterangeQuerytempend = daterangeQuerytemparray[1].substring(1)
        var daterangeQuery = "&tbs=cdr:1,cd_min:" + daterangeQuerytempstart + ",cd_max:" + daterangeQuerytempend
}
else {
        var daterangeQuery = ""
}

//检查语言参数
var regLanguageCN = /\s中文/
var regLanguageEN = /\s英文/
if (searchQuery.match(regLanguageCN)){
        var langQuery = "&lr=lang_zh-CN"
}
else if (searchQuery.match(regLanguageEN)){
        var langQuery = "&lr=lang_en"
}
else {
        var langQuery = ""
}

var pureQuery = searchQuery.replace(regDate,'').replace(regDaterange,'').replace(regLanguageCN,'').replace(regLanguageEN,'')

var searchURL = "https://www.google.hk/search?q=" + pureQuery + dateQuery + daterangeQuery + langQuery
let encodeURI_url = encodeURI(searchURL)
```

脚本简析

最后得到的**搜索 URL** 是关键，接着就可以依据所用的自动化工具，放到网页里面打开查看。在 Drafts 中我们可以后接一个「URL」步骤直接打开 URL。Drafts 内置了 Safari View Controller，能够在应用内直接预览搜索结果，你所获得的动作默认也是这个设置。当然，喜欢在浏览器中查看搜索结果也可以，只需要取消勾选下图的「Open in Drafts」选项。


Steps
Save & Exit
URL
TEMPLATE

[[jsResult]]

OPTIONS

Open in Drafts ●

URL encode tags ○

URL 步骤

至于「URL」步骤是如何从 JavaScript 获得搜索 URL 的，留到最后模板部分统一讲。

添加自定义结果语言

本文所提供的脚本只提供 `中文` 和 `英文` 两个搜索结果语言参数，理所当然，你可以添加其他的语言。

以把 `英文` 改成其他语种为例[5]

，首先找到 `//检查语言参数` 开始的这部分的代码（绿色部分），连着剔除自然语言参数的部分（蓝色部分），一共需要修改这几处：

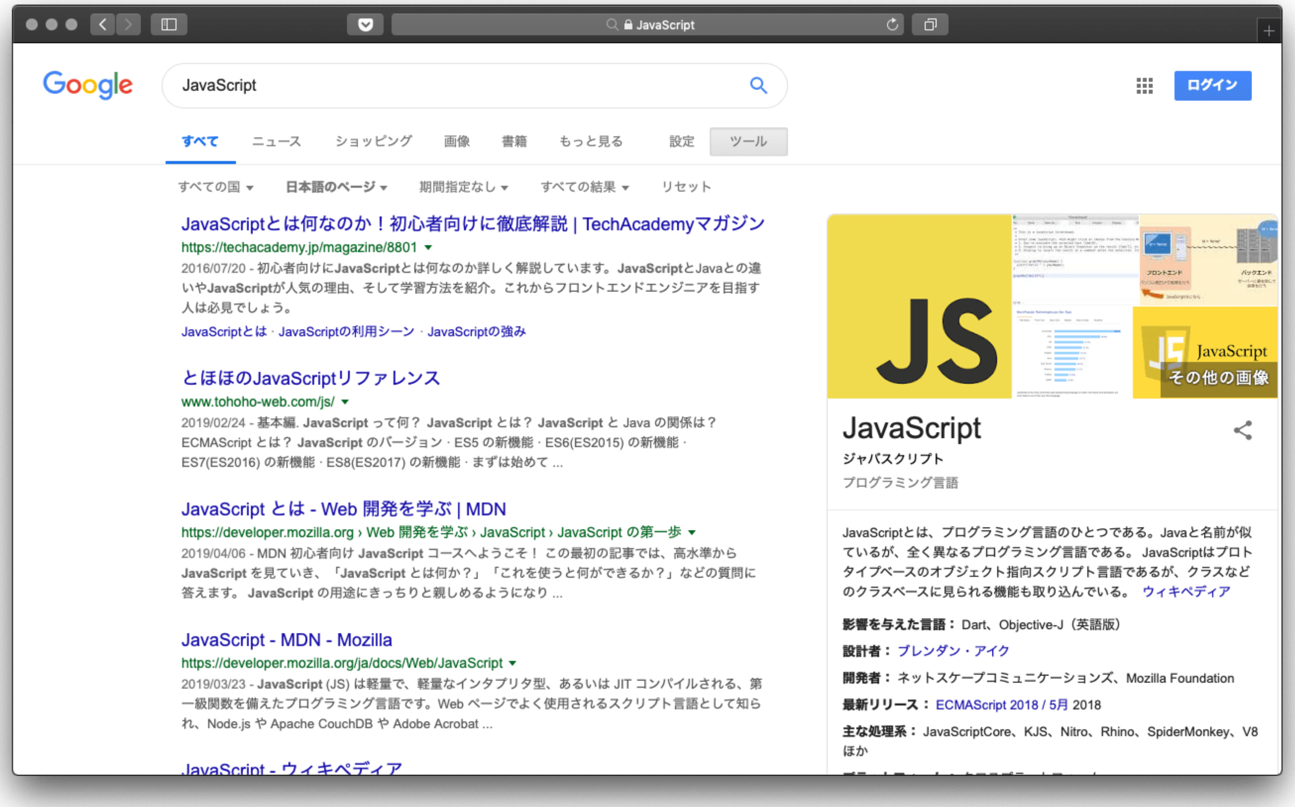
1. **修改结果语言参数的样式名称**：将**三处** `regLanguageEN` 改成 `自定义语言样式`，最后一处在删除自然语言参数的代码中（下面代码块的最后一行）。实际使用时尽量用英文，下面几条也同理。

2. **修改具体要查找的参数**：将 `英文` 改成 `自定义语种`。

3. **修改对应的 Google 搜索 URL 参数**：将 `lang_en` 改成所需语种的 URL 参数，具体可在 这里 找到。

```
//检查语言参数 var regLanguageCN = /\s中文/ var 自定义语言样
式 = /\s自定义语种/ if (searchQuery.match(regLanguageCN)){
var langQuery = “&lr=lang_zh-CN” } else if
(searchQuery.match(自定义语言样式)){ var langQuery = “&lr=
语言参数” } else { var langQuery = “” }

var pureQuery = searchQuery.replace(regDate,
‘’).replace(regDaterange,‘’).replace(regLanguageCN,
‘’).replace(自定义语言样式,‘’)
```

这样修改之后，就可以将搜索结果限定为自己需要的其他语言了。下图是一个仅检索日文内容的示例，其他语种的使用者也可以制作专用搜索参数。



JavaScriptとは何なのか！初心者向けに徹底解説 | TechAcademyマガジン
https://techacademy.jp/magazine/8801
2016/07/20 - 初心者向けに**JavaScript**とは何なのか詳しく解説しています。**JavaScript**とJavaとの違いや**JavaScript**が人気の理由、そして学習方法を紹介。これからフロントエンドエンジニアを目指す人は必見でしょう。
JavaScriptとは · JavaScriptの利用シーン · JavaScriptの強み

とほほのJavaScriptリファレンス
www.tohoho-web.com/js/
2019/02/24 - 基本編. **JavaScript** って何？ **JavaScript** とは？ **JavaScript** と Java の関係は？ ECMAScript とは？ **JavaScript** のバージョン · ES5 の新機能 · ES6(ES2015) の新機能 · ES7(ES2016) の新機能 · ES8(ES2017) の新機能 · まずは始めて ...

JavaScript とは - Web 開発を学ぶ | MDN
https://developer.mozilla.org › Web 開発を学ぶ › JavaScript › JavaScript の第一歩
2019/04/06 - MDN 初心者向け **JavaScript** コースへようこそ！ この最初の記事では、高水準から **JavaScript** を見ていき、「**JavaScript** とは何か？」「これを使うと何ができるか？」などの質問に答えます。 **JavaScript** の用途にきっちりと親しめるようになり ...

JavaScript - MDN - Mozilla
https://developer.mozilla.org/ja/docs/Web/JavaScript
2019/03/23 - **JavaScript** (JS) は軽量で、軽量なインタプリタ型、あるいは JIT コンパイルされる、第一級関数を備えたプログラミング言語です。Web ページでよく使用されるスクリプト言語として知られ、Node.js や Apache CouchDB や Adobe Acrobat ...

JavaScript - ウィキペディア

JavaScript
ジャバスクリプト
プログラミング言語

JavaScriptとは、プログラミング言語のひとつである。Javaと名前が似ているが、全く異なるプログラミング言語である。 JavaScriptはプロトタイプベースのオブジェクト指向スクリプト言語であるが、クラスなどのクラスベースに見られる機能も取り込んでいる。 ウィキペディア

影響を与えた言語： Dart、Objective-J（英語版）
設計者： ブレンダン・アイク
開発者： ネットスケープコミュニケーションズ、Mozilla Foundation
最新リリース： ECMAScript 2018 / 5月 2018
主なな処理系： JavaScriptCore、KJS、Nitro、Rhino、SpiderMonkey、V8ほか

限制搜索结果为日文

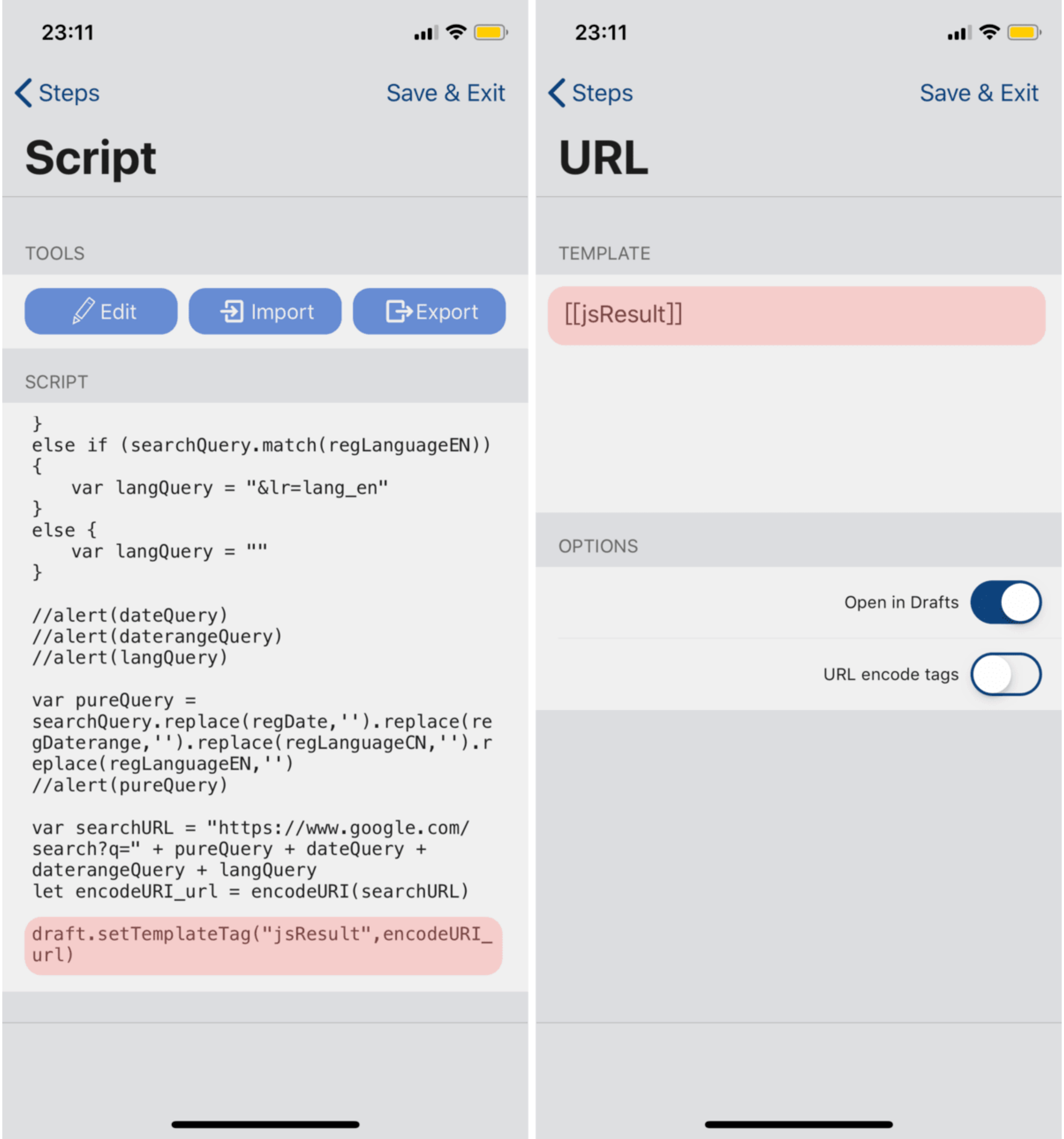
一个 Drafts 自动化模板套件

为求雨露均沾，前文讲到如何获取 JavaScript 步骤中的数据时就戛然而止，先转到了更通用的自定义语言部分。这里再给各位 Drafts 用户补上相关内容。

有必要说明一下，在 Drafts 的 JavaScript 步骤中也可以直接打开网页，但是完全靠代码来做事似乎有点浪费 Drafts 原生功能的发挥空间，何况对不熟悉 JavaScript 的读者也不友好。我推荐的方案，则是结合 **JavaScript** 和 Drafts 的**标签**，既能保留脚本的强大功能，又可以在 URL 等步骤中通过标签方便地使用各种数据[6]

。下面红色高亮部分就是一个泛用的 **JavaScript&标签**模

板。


23:11
Steps
Save & Exit
Script
TOOLS
Edit
Import
Export
SCRIPT
23:11
Steps
Save & Exit
URL
TEMPLATE
OPTIONS

```
}
else if (searchQuery.match(regLanguageEN))
{
    var langQuery = "&lr=lang_en"
}
else {
    var langQuery = ""
}

//alert(dateQuery)
//alert(daterangeQuery)
//alert(langQuery)

var pureQuery =
searchQuery.replace(regDate,'').replace(re
gDaterange,'').replace(regLanguageCN,'').r
eplace(regLanguageEN,'')
//alert(pureQuery)

var searchURL = "https://www.google.com/
search?q=" + pureQuery + dateQuery +
daterangeQuery + langQuery
let encodeURI_url = encodeURI(searchURL)
draft.setTemplateTag("jsResult",encodeURI_
url)
```

[[jsResult]]

Open in Drafts ●
URL encode tags ○

从 JavaScript 脚本传递数据出去的模板

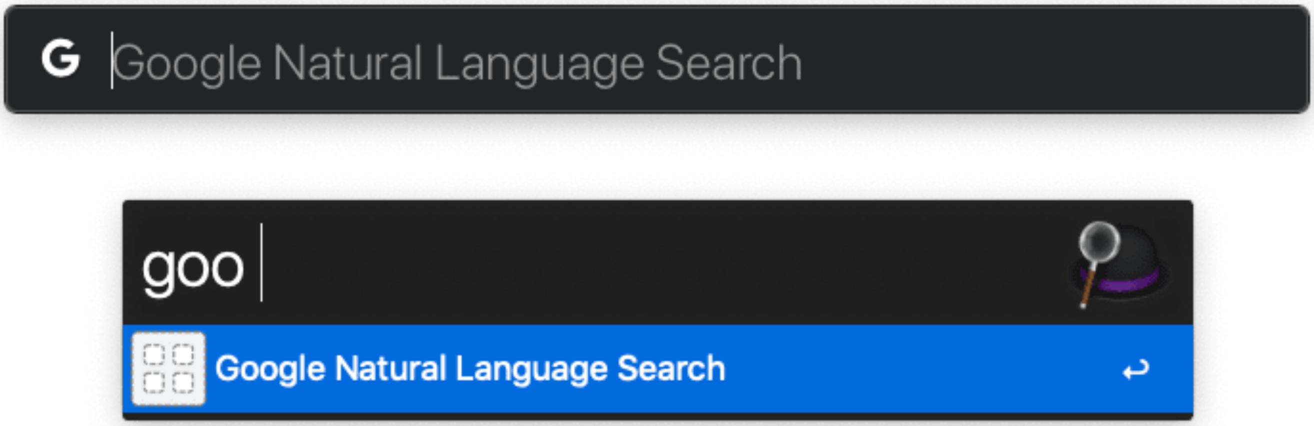
Drafts 里的标签都是 `[[标签名]]` 的样式，除了自带的草稿标题 `[[title]]`、内容 `[[content]]`，还可以自己设置标签。在 JavaScript 脚本末尾加入上图左侧末尾的代码，就可以创建一个标签供本动作后续步骤使用（不会影响其他的动作）：

```
draft.setTemplateTag("标签名","标签内容")
```

在「URL」步骤中我们可以看到 `[[jsResult]]` 这一标签，通过它就能获得整合出来的 Google 搜索 URL。从分析输入的关键词到最后的搜索，整个过程就此完成。

Mac 端实现和注意事项

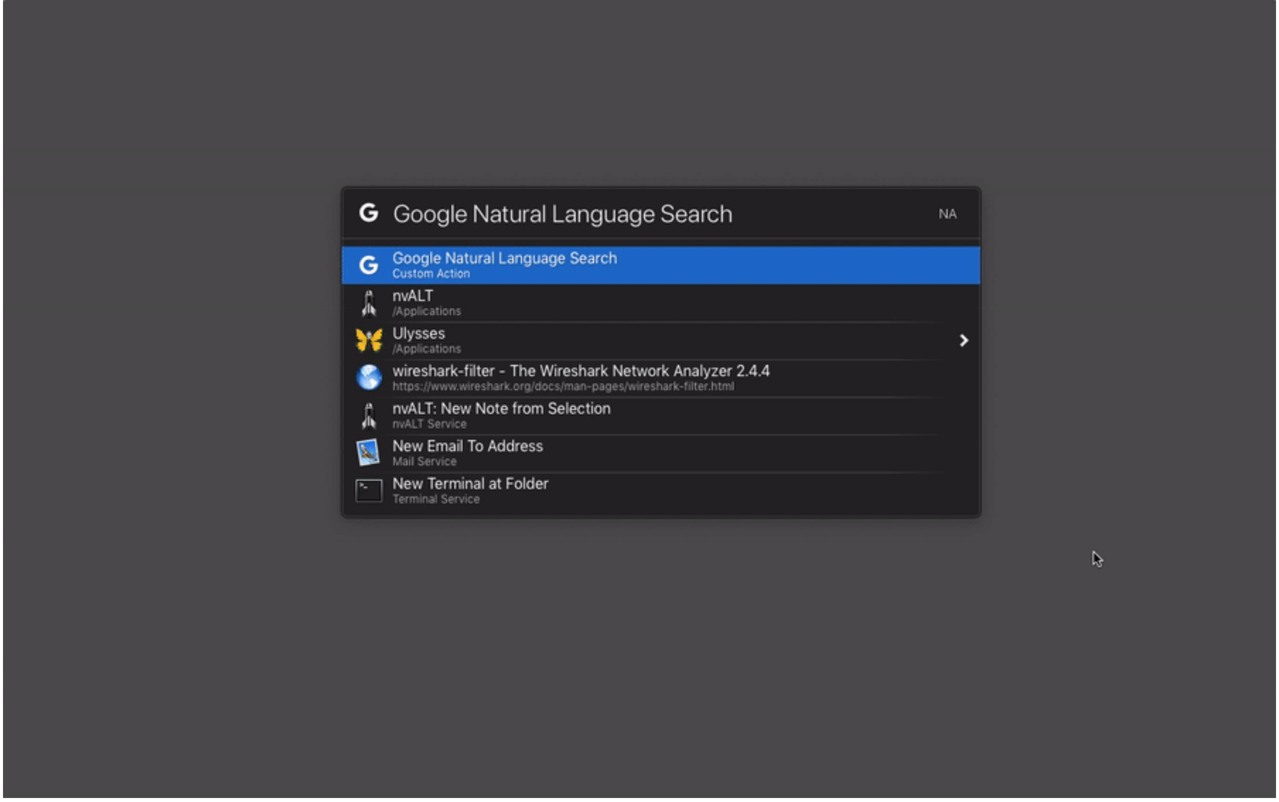
由于搜索脚本完全用 JavaScript 写成，理论上任何支持 js 的工具都可以使用。不过最理想的工具载体还是**各种启动器**，LaunchBar、Alfred 这些启动工具本身就是一个输入框，配合自然语言工作起来严丝合缝。


Google Natural Language Search
goo
Google Natural Language Search

启动器和自然语言搜索配合最舒服

LaunchBar 和 Alfred 两版的操作略有不同，这里还是简单介绍一下：

1. **LaunchBar 版**：输入「Google Natural Language Search」的前几个字母，直到出现这一动作为止，按下空格继续按语法输入关键词和参数来搜索。



Google Natural Language Search NA

Google Natural Language Search
Custom Action

nvALT
/Applications

Ulysses
/Applications

wireshark-filter - The Wireshark Network Analyzer 2.4.4
https://www.wireshark.org/docs/man-pages/wireshark-filter.html

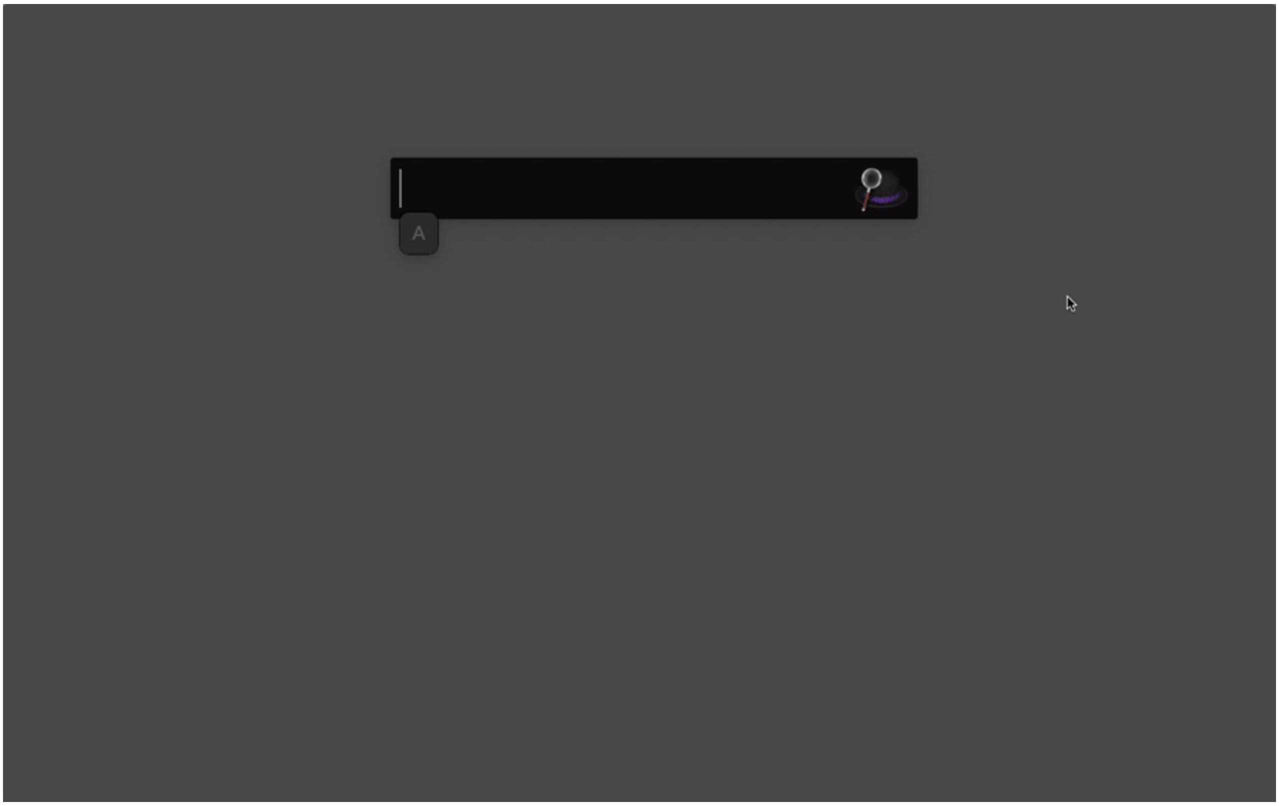
nvALT: New Note from Selection
nvALT Service

New Email To Address
Mail Service

New Terminal at Folder
Terminal Service

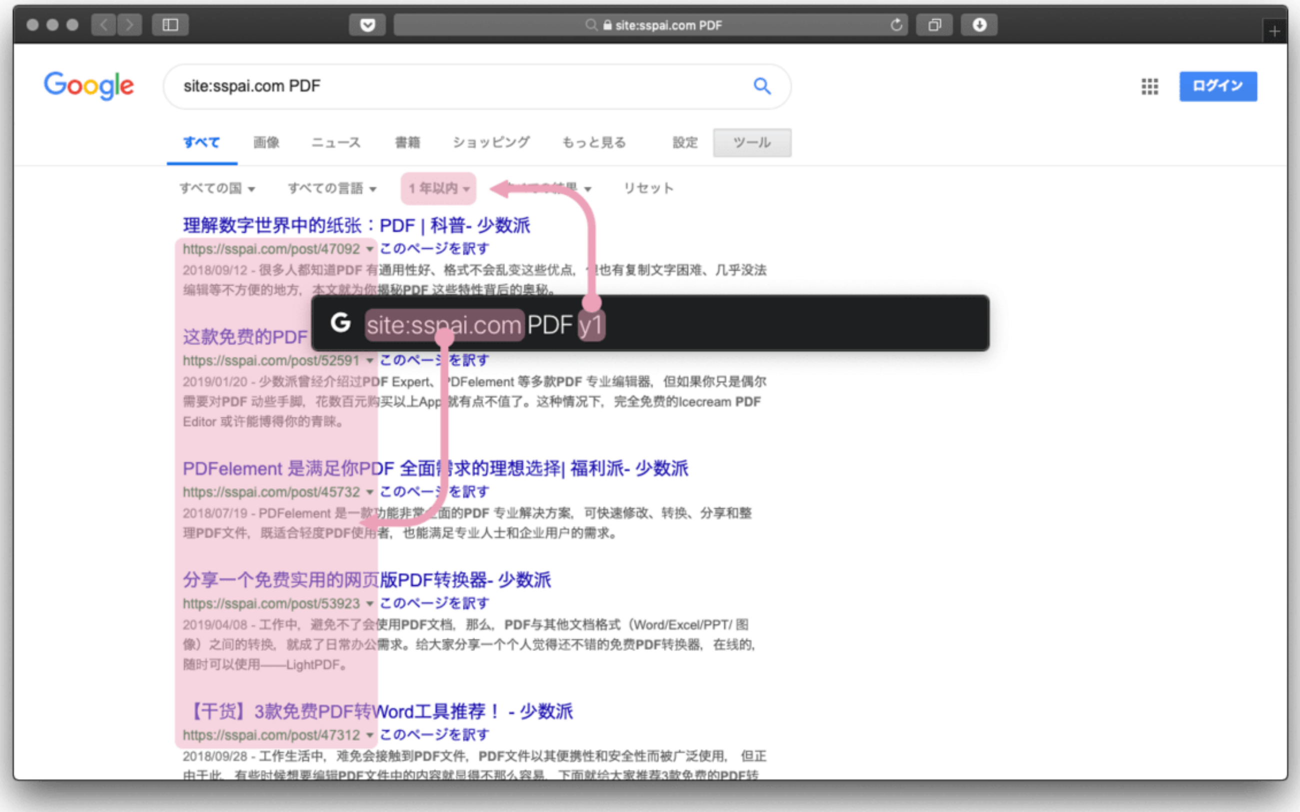
在 LaunchBar 中用自然语言搜索

2. **Alfred 版**：输入 goo 关键词来显示「Google Natural Language Search」工作流（workflow），空一格后输入关键词和参数，按下回车即可搜索。



在 Alfred 中使用自然语言进行 Google 搜索

使用时需要注意一个细节，在没开启 Safari 时，LaunchBar 或 Alfred 会先停顿一下来打开浏览器，接着再开启搜索页面。如果遇到搜索框有几秒钟没反应，一般就是浏览器没打开，并不是卡顿或出错。Mac 上的自然语言搜索整体功能和 Drafts 保持一致，不过在 Mac 上可以随时呼出搜索框，这种有人在身边待命的感觉更舒服些。



site:sspai.com PDF

site:sspai.com PDF y1

理解数字世界中的纸张：PDF | 科普- 少数派
https://sspai.com/post/47092 このページを訳す
2018/09/12 - 很多人都知道PDF 有通用性好、格式不会乱变这些优点，也有复制文字困难、几乎没法编辑等不方便的地方，本文就为你揭秘PDF 这些特性背后的奥秘。

这款免费的PDF
https://sspai.com/post/52591 このページを訳す
2019/01/20 - 少数派曾经介绍过PDF Expert、DFelement 等多款PDF 专业编辑器，但如果你只是偶尔需要对PDF 动些手脚，花数百元购买以上App 就有点不值了。这种情况下，完全免费的Icecream PDF Editor 或许能博得你的青睐。

PDFelement 是满足你PDF 全面需求的理想选择| 福利派- 少数派
https://sspai.com/post/45732 このページを訳す
2018/07/19 - PDFelement 是一款功能非常全面的PDF 专业解决方案，可快速修改、转换、分享和整理PDF文件，既适合轻度PDF使用者，也能满足专业人士和企业用户的需求。

分享一个免费实用的网页版PDF转换器- 少数派
https://sspai.com/post/53923 このページを訳す
2019/04/08 - 工作中，避免不了会使用PDF文档，那么，PDF与其他文档格式（Word/Excel/PPT/ 图像）之间的转换，就成了日常办公需求。给大家分享一个个人觉得还不错的免费PDF转换器，在线的，随时可以使用——LightPDF。

【干货】3款免费PDF转Word工具推荐！ - 少数派
https://sspai.com/post/47312 このページを訳す
2018/09/28 - 工作生活中，难免会接触到PDF文件，PDF文件以其便携性和安全性而被广泛使用，但正

搜索语法对于的过滤条件

设置专用域名后缀

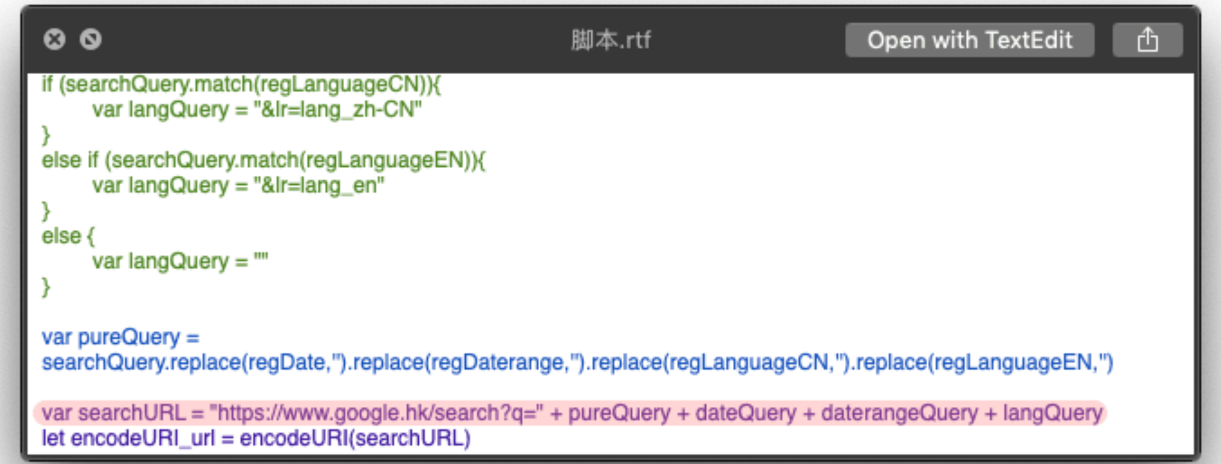
两个版本的动作代码都差不多，就不再重复分析，需要自定义结果语言范围的读者可以参考 Drafts 部分。下面，我们专注一个 Mac 端的特殊情况：**根据所处网络，为搜索动作设置对应地区**。

在 Drafts 版的动作中，最终搜索 URL 的开头就是常见的 Google 搜索前半部分：

```
https://www.google.com
```

一般来说，不管使用者身处哪个国家，都会跳转到预期的搜索结果页面。但是电脑上可能不太一样，视所处国家或网络状况需不同，搜索 URL 的域名可能要改动，比如在香港的朋友就需要将 `.com` 改成 `.hk`，**不然自然语言参数可能会失效。** 就拿香港地区为例，脚本生成搜索 URL 部分的第一行（即红色高亮部分）需要修改为：

```
var searchURL = "https://www.google.hk/search?q=" + pureQuery + dateQuery + daterangeQuery + langQuery
```



```
if (searchQuery.match(regLanguageCN)){
	var langQuery = "&lr=lang_zh-CN"
}
else if (searchQuery.match(regLanguageEN)){
	var langQuery = "&lr=lang_en"
}
else {
	var langQuery = ""
}

var pureQuery = 
searchQuery.replace(regDate,'').replace(regDaterange,'').replace(regLanguageCN,'').replace(regLanguageEN,'')

var searchURL = "https://www.google.hk/search?q=" + pureQuery + dateQuery + daterangeQuery + langQuery
let encodeURI_url = encodeURI(searchURL)
```

修改搜索 URL

处于其他地区和网络环境下的读者，遇到自然语言参数无效

时也可以这样修改。

自定义默认浏览器

另一个可以自己修改的地方是**默认浏览器**。两个动作所用的代码都是嵌在 JXA 中的标准 JavaScript，可以通过苹果的原生语法调用一部分软件控件，打开浏览器网页就用到了 JXA 语法：

```
var Safari = Application("Safari");
var doc = Safari.Document().make();
var win = Safari.windows[doc.name()];
win.currentTab.url = encodeURI_url
```

如果你想把浏览器改成别的，比如 Chrome，可以这么改：

```
var Chrome = Application("Chrome");
var doc = Chrome.Window().make();
var win = Chrome.windows[doc.name()];
win.currentTab.url = encodeURI_url
```

至于更多浏览器的用户，这里没法一一照顾过来，有需要的读者可以参考 这篇文章 中收录的语法细节。

此外，由于 Alfred 对于输入数据的处理和 LaunchBar 不同，所以你在脚本开头会看到一行专门的代码，它的作用是把输入的数组转换为普通字符串：

```
var searchQuery = argv.join()
```

不熟悉它们也无妨，不影响正常使用动作，不过有兴趣自己制作 Alfred workflow 的读者可以留意一下。

小结

前几周恰逢 Google I/O 大会，媒体报道基本是些通稿，甚至还有炒前几年冷饭的；为了获得更精确的搜索结果，我每天起床都搜一下 `google I/O d1`，便知道最近一天大会上大致又发布了哪些软硬件和服务。

这只是所谓「自然语言搜索」的一个小用例。在其他需要精确日期、时间或语言的场合，这一搜索方式同样能够发挥作用。